IIC3432 — Advanced topics in databases

Homework 3 projects

1. Projects

A representative of the group should send me an email ccing the other person in the group with two preferences for a project they would like to do. It is recommendable that you do this before Monday 23rd of October at 15pm to get the topic you wish to work on. One topic can not be taken by two groups.

Presentations will be held on the 2nd of November.

1.1. MongoDB vs other document databases

This project is about benchmarking different document databases. In particular, you should study the benchmark available at `https://www.arangodb.com/2015/10/benchmark-postgresql-mongodb-arangodb/` and pit MongoDB against another document database (e.g. ArangoDB) over this benchmark. Since the benchmark was made to make Arango look good, you must also test how the two systems you selected work over the wikidata dataset (the big one) we used in class. In particular, design 10 queries over wikidata and test how they perform in the two databases. In your presentation you should explain what is the data/queries being used to test the two systems.

1.2. Indices in MongoDB

Quite simple: explain the basic types of indices available in MongoDB, what they are, what they enable you to do, and how are they realized in Mongo. You should cover at least: single field index (including embedded field and embedded document), compound index, multikey index and (text or geospatial) index. You should also obtain/create a dataset that allows you to demonstrate the utility of an index in EACH of the cases above, showing us what is won when an index is present, and how the query performs when there is no index. Your dataset should be of reasonable size, so that the running times are at least in seconds and not only in miliseconds.

1.3. GraphQL

Facebook has a nice proposal of how APIs should be created/treated/maintained called GraphQL. The documentation of the project is available at: `http://graphql.org/` and `http://facebook.github.io/graphql`. Your task here is to understand and explain what

GraphQL does, and more importantly, to understand how does this proposal relate to the database paradigms we discussed in this course (hint: graph and document ones play a role). There is also a paper which might be of help (but it might also confuse you) available at `http://ceur-ws.org/Vol-1912/paper11.pdf`. In the presentation you should explain what the data model/query language behind the project is, and give us live examples how it works (either through a live API, or through one you install locally).

1.4. MongoDB vs SQL

Find two benchmarks: one relational, and another one made for NoSQL/document databases. There are lots of such benchmarks available on-line, most famous being: TPC-H, Berlin (relational), or EndPoint, YCSB (noSQL); plus most noSQL databases boast about how they beat the competition on their respective Web sites (e.g Arango). In this task you are meant to measure how the two technologies compare in a single-node (i.e. not distributed) setting. You should select two benchmarks (relational + noSQL), and compare a SQL and a document/key-value based one on the two benchmarks. You should explain what the two benchmarks do, how you represented the data in the opposite model, and what results you got. You are allowed to try and add indices in both settings to try to boast performance (but are not required to do so).

Note 1: if some benchmark is too big for your machine, you can use only a part of it.
Note 2: you do not need to select any of the benchmarks I proposed – it is up to you to select which ones you wish to work with.

1.5. Testing CAP compliance

In class we explained that a distributed system can guarantee at most two of the three CAP constraints at a given time, or that it can guarantee some mid-point where it never really complies with all three, but is quite close (the BASE paradigm). In this project, your task is to verify how document or key-value stores comply with their specification of this guarantee. More precisely, you should select one document or key-value store (e.g. Mongo, Arango, Cassandra, or CouchDB), set up multiple virtual nodes, and test what happens when there is a network partition. You should then chose to check how the systems enforces either consistency (e.g. in Mongo), or availability + eventual consistency (e.g. in Cassandra), or some other guarantee that the system gives you. Your simulation should show us how this happens in practice.

1.6. Benefits and drawbacks of a distributed setting

In this project you are supposed to gauge the benefits and drawbacks of having a distributed (partially) replicated database spread over multiple nodes. To do this, you should select one noSQL system (e.g. Mongo, Arango, Cassandra, or CouchDB), figure out how to set up virtual nodes, and gauge the performance of the system depending on the number of nodes you

set up. Some nice datasets/benchmarks to test this are YCSB or some of these: `https://academy.datastax.com/planet-cassandra/nosql-performance-benchmarks`. Check basic properties (e.g. read/write perfomance) and see how they improve with multiple nodes. As for drawbacks, consider the cost of this (storage, replication, etc.).

Note: if some benchmark is too big for your machine, you can use only a part of it.

1.7. Propose your own topic

Quite simple: send me an email (dvrgoc@ing.puc.cl) to confirm the topic you would like to explore.